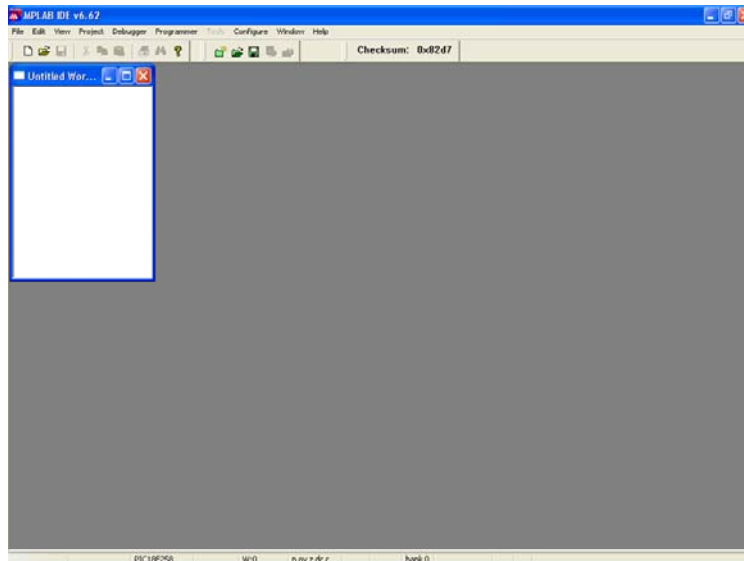


ANEXO I

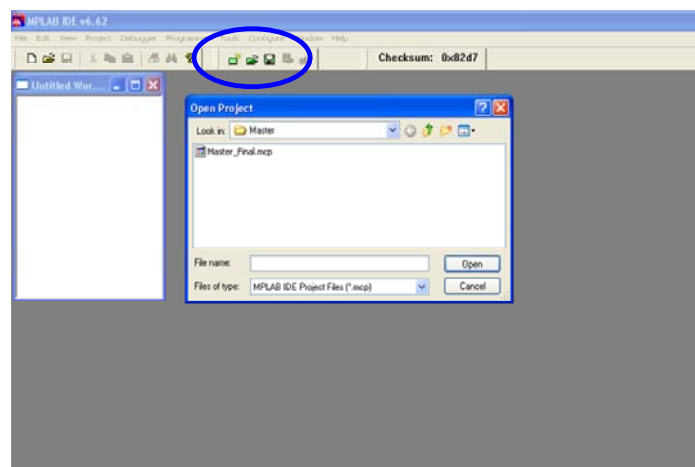
O Ambiente de desenvolvimento MpLab IDE v6.62

Como ferramenta de desenvolvimento do código dos microcontroladores utilizou-se o MpLab IDE v6.62 da Microchip. A linguagem de programação utilizada foi o C, como compilador utilizou-se o MCC18 da Microchip. Este compilador permite ser integrado no ambiente de desenvolvimento.

Aspecto gráfico do ambiente de desenvolvimento do MpLab.



Anexo - Figura 1

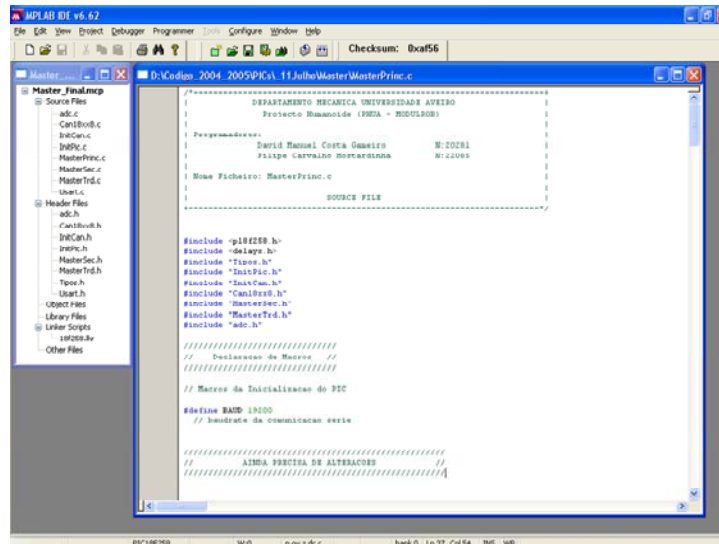


Anexo - Figura 2

A seguir será descrito o processo necessário para abrir um projecto existente.

Para abrir um projecto existente basta escolher o botão com a pasta verde na barra de ferramentas. Para abrir ficheiros sem que se queira abrir todo o projecto utiliza-se o outro botão com a pasta amarela.

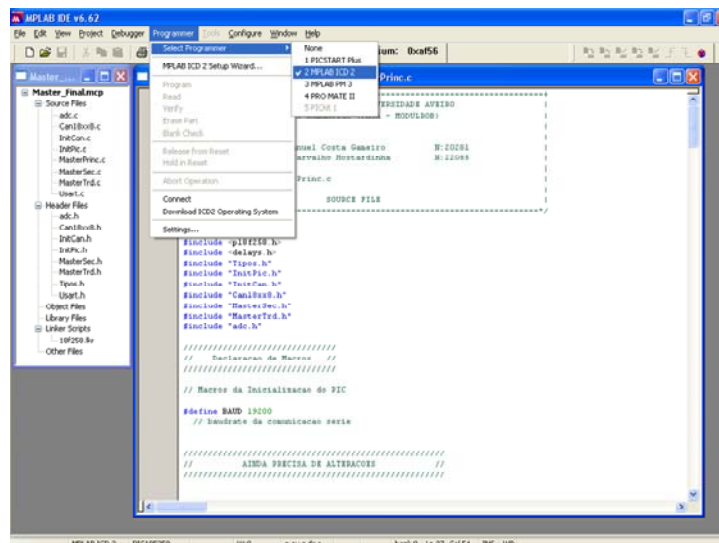
Ao abrir um projecto temos acesso a todos os ficheiros que fazem parte do projecto.



Anexo - Figura 3

A seguir será descrito o processo necessário para programar um microcontrolador utilizando o código de um projecto desenvolvido utilizando o programador da MpLab ICD2.

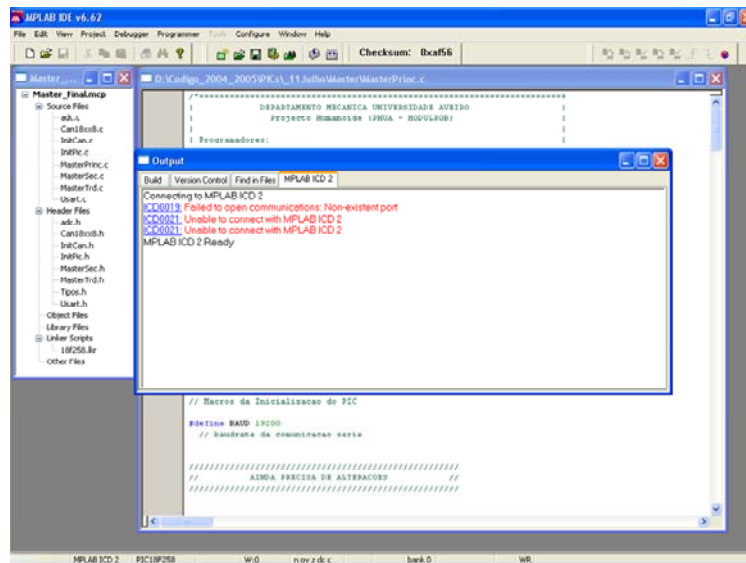
Na barra de menus do MpLab existe o menu Programmer. Neste menu escolhe-se o programador que queremos utilizar, neste caso o MpLab ICD2.



Anexo - Figura 4

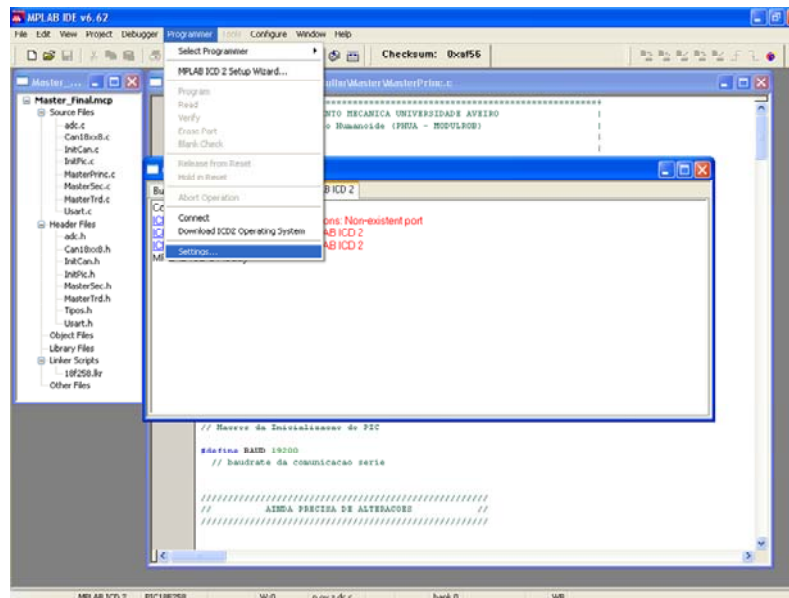
Como se pode ver na figura aparece uma nova janela designada por Output. Esta janela é muito importante pois todas as mensagens que o MpLab devolve para o utilizador são

apresentadas nesta janela. Como se pode por exemplo ver na figura são devolvidas mensagens de erro ao ligar-se pela primeira vez o ICD2.



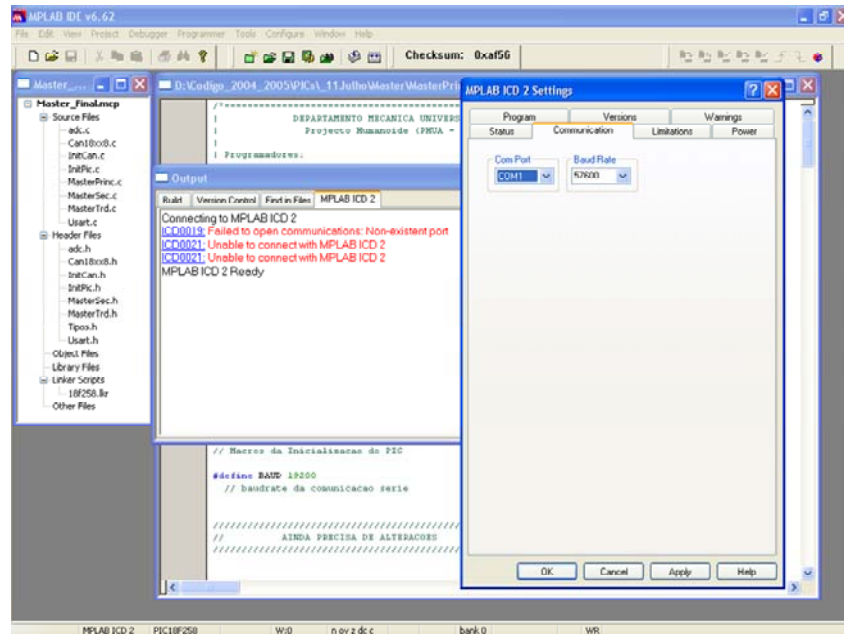
Anexo - Figura 5

As mensagens indicam que não foi possível comunicar com o ICD2. Para estabelecer a ligação correcta é necessário escolher o submenu Settings do menu Programmer.

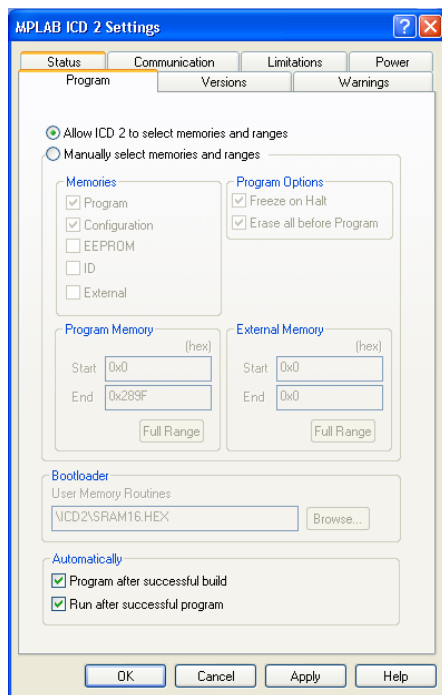


Anexo - Figura 6

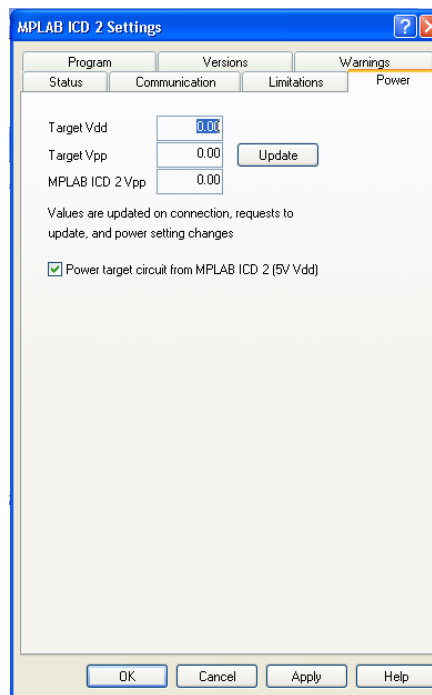
Dentro deste submenu de configurações é necessário o frame Communication. Aqui pode-se escolher o Port de comunicações e a baud rate a que se comunica. A comunicação por USB nunca foi utilizada pelo qual se aconselha a utilização da comunicação Série.



Anexo - Figura 7



Anexo - Figura 8



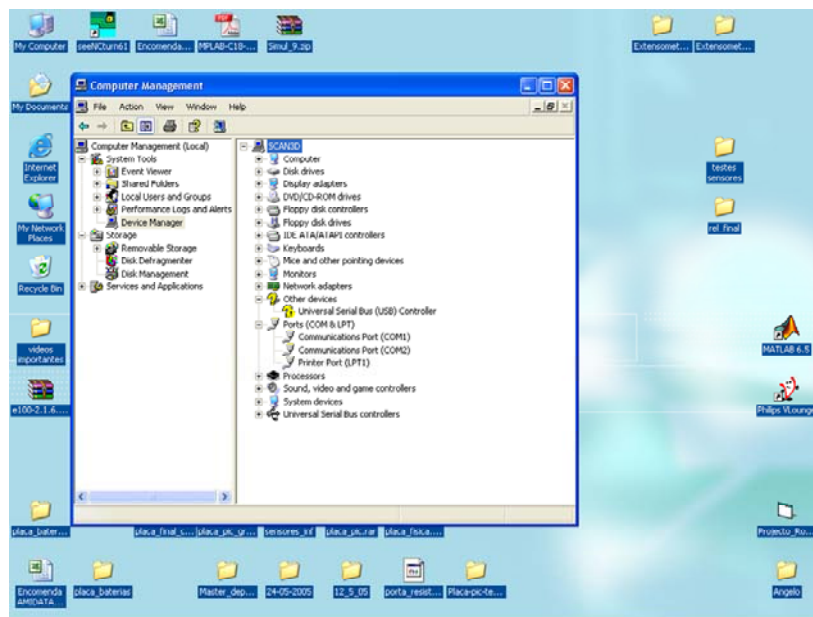
Anexo - Figura 9

Nas restantes configurações são utilizadas as opções originais. Basta então carregar em Apply e OK. Ao efectuar se isto o MpLAB devolve o seguinte aviso.



Anexo - Figura 10

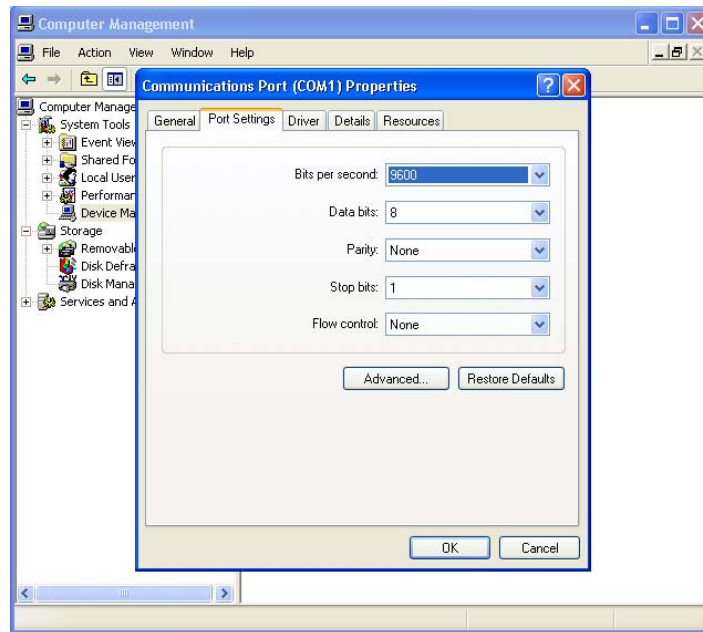
O Windows tem por defeito os FIFO buffers da comunicação série activados. Temos que ir ao My Computer desctivar os serial FIFO buffers. A maneira mais rápida é carregar com a tecla direita do rato sobre o icon do meu computador. Ao efectuar-se isto aparece o menu apresentado na figura. Escolhe-se o submenu Manage.



Anexo - Figura 11

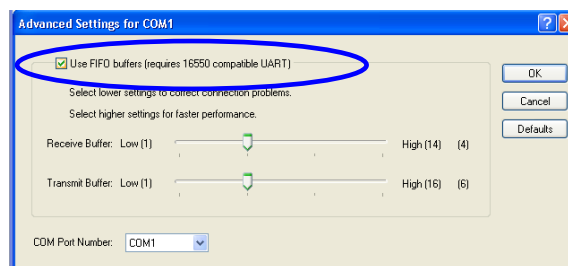
Nesta janela escolhe-se o dispositivo de comunicação série que se pretende alterar a comunicação.

Escolhe-se na nova janela o frame Port Setting. E selecciona-se o botão Advanced.



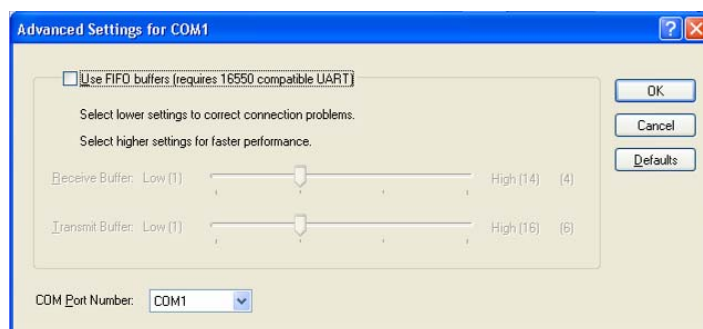
Anexo - Figura 12

Na janela Advanced Settings desactiva-se o FIFO buffers da Porta de comunicações.



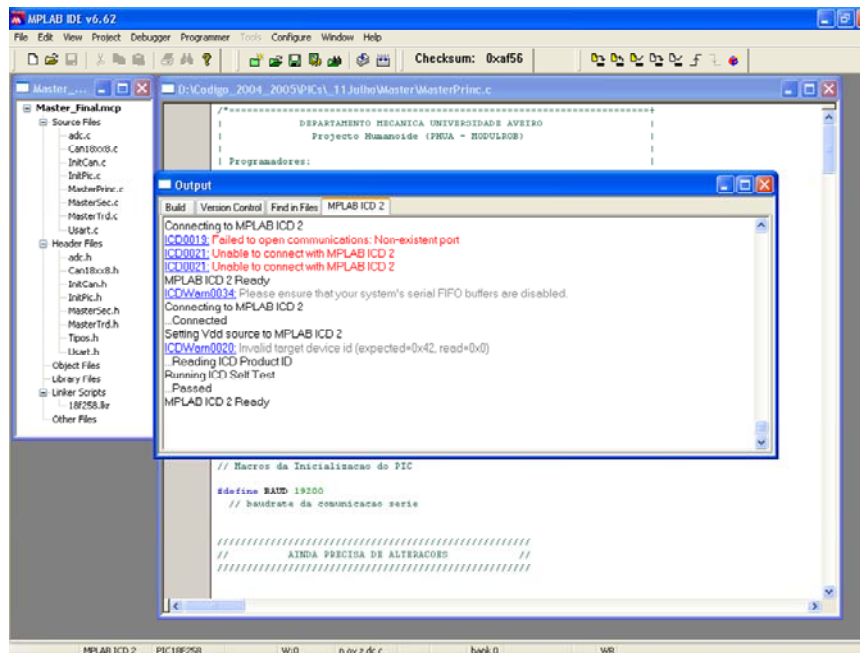
Anexo - Figura 13

Neste momento o MpLab já deve de ser capaz de estabelecer comunicação com o ICD2.



Anexo - Figura 14

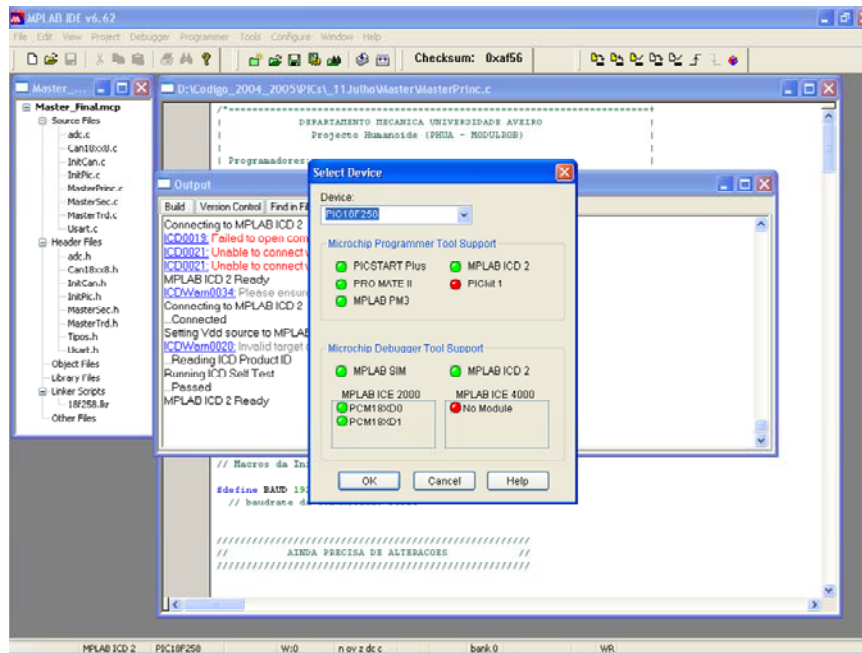
Como é possível verificar as comunicações foram estabelecidas correctamente.



Anexo - Figura 15

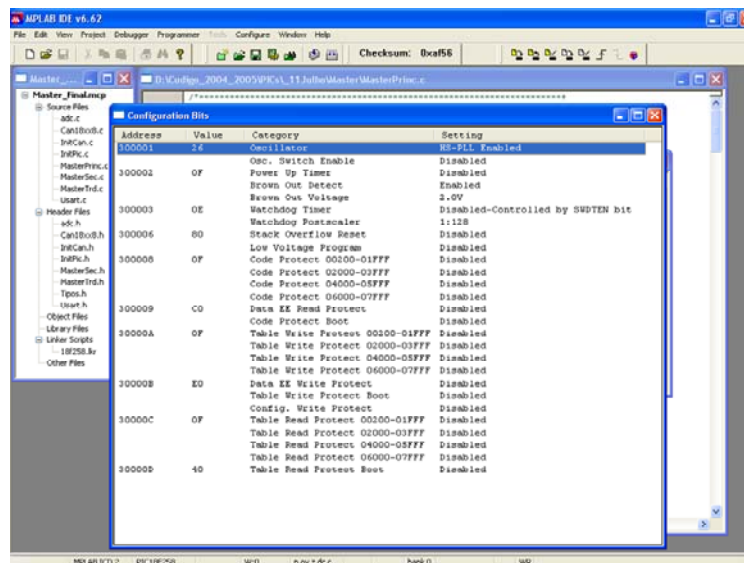
No entanto continua a aparecer um Warning a indicar que o ID do microcontrolador não é válido. Este Warning pode várias ter várias causa. A primeira a indicar será a falta de alimentação do microcontrolador. Verificar a placa que se está utilizar e confirmar se a montagem está de acordo com o esquema que se encontra na documentação do ICD2.

A segunda causa pode ser um problema de configuração. Sempre que se cria um projecto temos que escolher para qual microcontrolador é destinado o código. Caso isso não se tenha feito é possível voltar escolher o microcontrolador que se pretende programar. Para isso tem-se que escolher o menu Configure e no sub-menu Select Device escolher o microcontrolador que se está a utilizar.



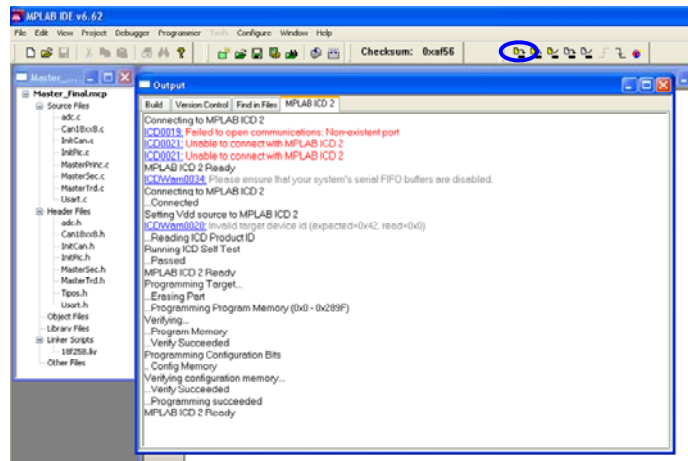
Anexo - Figura 16

Para além disto é necessário seleccionar os bits de configuração do microcontrolador. Apresenta-se na figura a configuração correcta.



Anexo - Figura 17

Após este procedimento deve se ter todo o sistema configurado e as comunicações devem estar estabelecidas. Podendo se compilar o código e programar o pic utilizando o botão que se indica na figura.

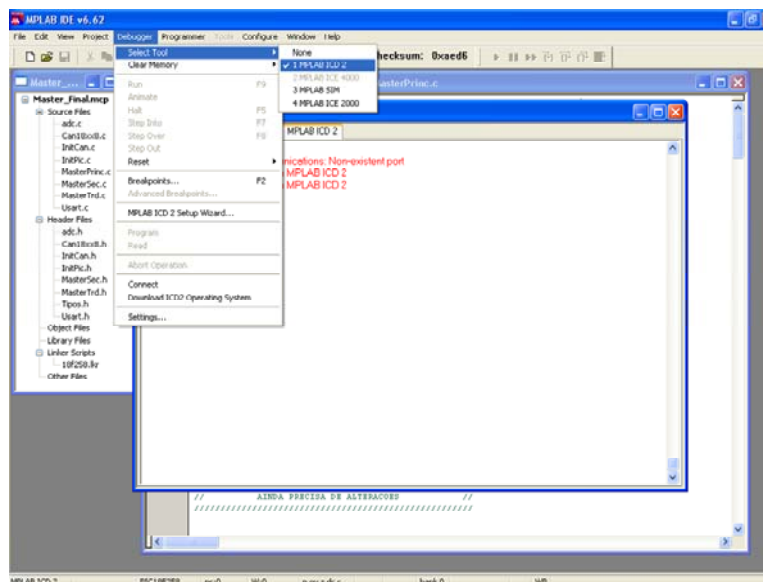


Anexo - Figura 18

A seguir será descrito o processo necessário para utilizar o ICD2 para efectuar o debugging de código a correr no microcontrolador.

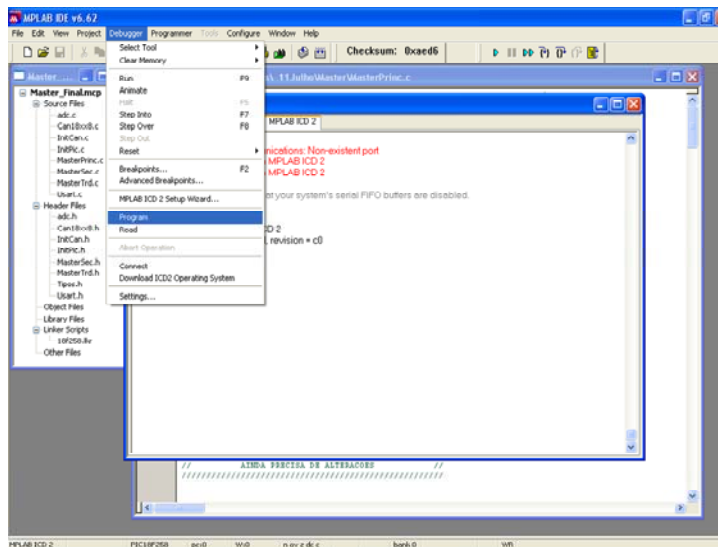
Na barra de Menus do MpLab existe o menu Debugger. No submenu Select Tool escolhemos o MpLab ICD2.

Caso o MpLab não reconheça o ICD2 tem-se que efectuar o procedimento descrito anteriormente.



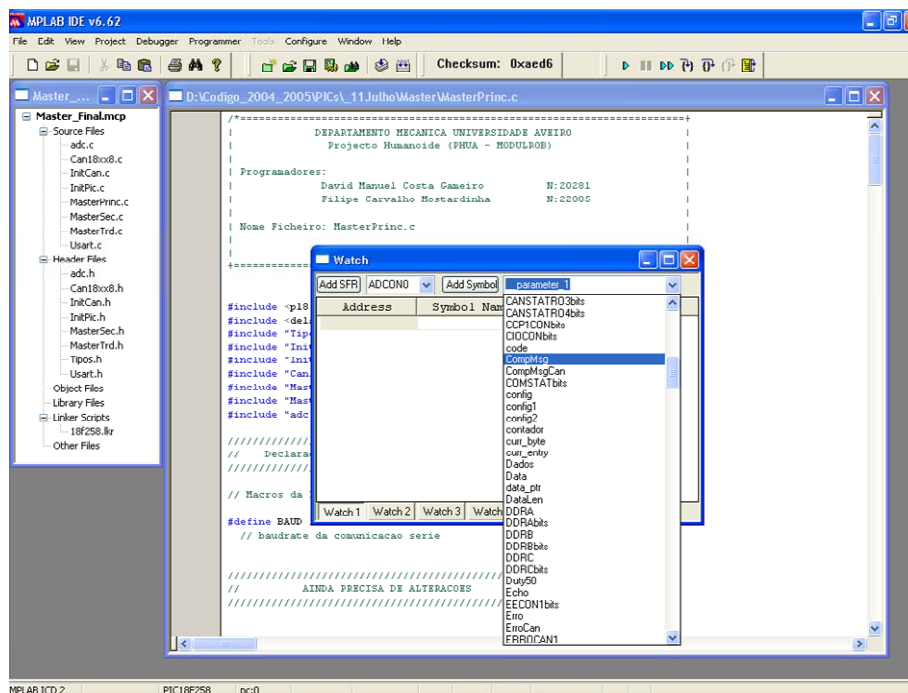
Anexo - Figura 19

Para o In Circuit Debugger (ICD) funcione é necessária uma rotina específica que é programada em conjunto com o código que se desenvolveu no projecto. Esta rotina serve para entre outras coisas controlar todo o funcionamento do microcontrolador em modo de debugging. O MpLab encarrega-se de efectuar isto por nós. Basta escolher a opção Program no menu Debugger que o MpLab programa o microcontrolador com tudo o que é necessário.

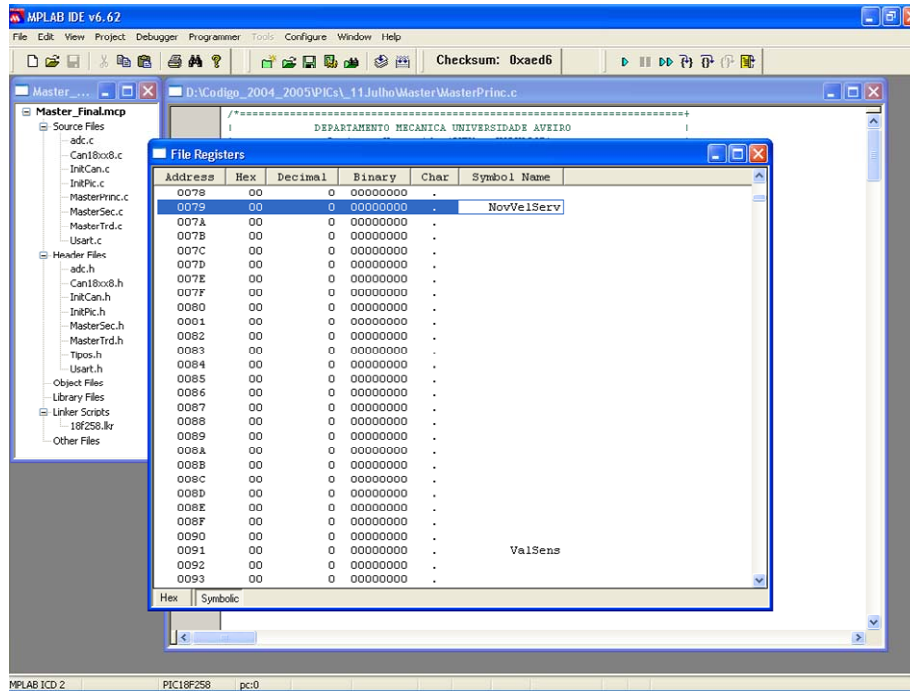


Anexo - Figura 20

Uma vez escolhido o debugger e programado o código aparece a barra de ferramentas do debugger. Aqui pode se colocar em ou interromper o funcionamento do microcontrolador para analisar o estado das variáveis e da memória na altura da interrupção. Basta no menu View escolher o submenu Watch ou File Register para se visualizar todas as variáveis do código do microcontrolador.

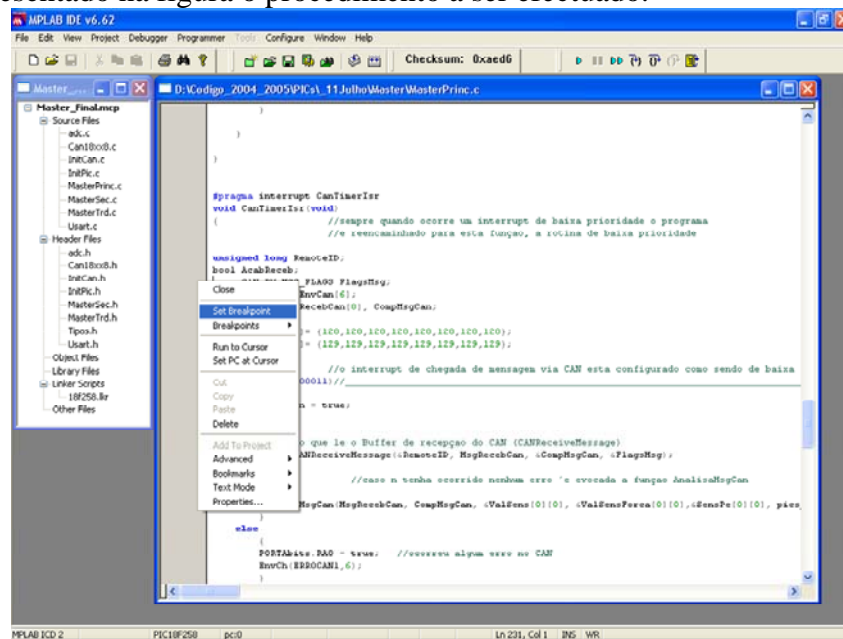


Anexo - Figura 21



Anexo - Figura 22

Neste ambiente de desenvolvimento existe também a possibilidade de colocar break points no código. É apresentado na figura o procedimento a ser efectuado.



Anexo - Figura 23

```
MPLAB IDE v6.62
File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0xacd6
D:\Codigo_2004_2005\PICa_11Julho\MasterMasterPrinc.c

Source Files
- ad.c
- Can18v0b.c
- InmCan.c
- InmPic.c
- MasterPrinc.c
- MasterSec.c
- MasterTrd.c
- Usart.c
Header Files
- ad.h
- Can18v0b.h
- InmCan.h
- InmPic.h
- MasterSec.h
- MasterTrd.h
- Tpo.h
- Usart.h
Object Files
Library Files
- I8255.lib
Other Files

)
}
}

#pragma interrupt CanTimerIsr
void CanTimerIsr(void)
{
    //sempre quando ocorre um interrupt de baixa prioridade o programa
    //e reencaminhado para esta funcao, a rotina de baixa prioridade

    unsigned long RemoteID;
    bool AcabReceb;
    bool CAN_RX_HDQ_PLASO FlagsMsg;
    static byte MsgRecebCan[6];
    static byte CompMsgCan;
    bool AcabEnv;
    byte ESDOCAN1[6] = {120,120,120,120,120,120};
    byte ESDOCAN2[6] = {129,129,129,129,129,129};

    //o interrupt de chegada de mensagens via CAN esta configurado como sendo de baixa
    if (PIEB = 0x00000011)
    {
        bool ErroCan = true;

        //funcao que le o Buffer de recepcao do CAN (CANReceiveMessage)
        ErroCan = CANReceiveMessage(&RemoteID, &MsgRecebCan, &CompMsgCan, &FlagsMsg);

        if (ErroCan)
            // caso n tenha ocorrido nenhum erro e ocorreu a funcao AnalisaMsgCan
            {
                AnalisaMsgCan(MsgRecebCan, CompMsgCan, &ValFena[0][0], &ValFenaFerea[0][0], &FenaFe[0][0], &Fena;
            }
        else
            {
                PORTA<u>L</u>.RA0 = true; //ocorreu algum erro no CAN
                EnvCh(ESDOCAN1, 6);
            }
    }
}

MPLAB ICD 2 PIC18F250 pc0 In 231, Col 1 INE WR
```

Anexo - Figura 24